Parallel Iterative Block and Direct Block Methods for 2-Space Dimension Problems on Distributed Memory Architecture

NORMA ALIAS, TAN YIN SAN, ROZIHA DARWIS, NORIZA SATAM, ZARITH SAFIZA A. GHAFFAR, NORHAFIZA HAMZAH, MD RAJIBUL ISLAM
Institute of Ibnu Sina, Department of Mathematics, Faculty of Science,
Universiti Teknologi Malaysia,
81310 Skudai, Johor, Malaysia
norma@ibnusina.utm.my, ibnu.qurra.iis@gmail.com

Abstract - *In numerical simulations of partial differential equations, it is often the case that we have to solve the matrix equations accrued from finite difference models of the equations. For computational purposes, we can iterate the solution system in such a way that the resulting matrices on the left hand side become easy to handle such as diagonal matrices or small matrices, for example the block systems. This indicates that we can apply various group computational molecules to simulate the partial differential equations numerically. In this paper, we present two problems of group schemes, specifically the Alternating Group Explicit (AGE) method and the Crack Propagation. We offer reasonable assessments and contrasts on behalf of the numerical experiments of these two methods ported to run through Parallel Virtual Machine (PVM) on distributed memory architecture.*

Keywords: *Alternating Group Explicit (AGE) Method, Element Stiffness method, Parallel Performance measurement, Distributed Memory Architecture, Parallel Virtual Machine (PVM).*

1 Introduction

1.1 Diffusion equation of parabolic PDE

We notice that an approximation to parabolic partial differential equations can be derived from the standard five-point finite difference approximation (Dahlquist & Bjorck 1974) to provide the two-dimensional diffusion partial differential equation,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + F(x,y,t) \tag{1}$$

with a specified initial and boundary conditions on a unit. One commonly used implicit finite difference scheme based on the centred difference in time and space formulation about the point $(i,j,k+\frac{1}{2})$ transforms equation (1) into

$$\begin{aligned} &-\lambda\theta u_{i-1,j,k+1} + (1+4\lambda\theta)u_{i,j,k+1} - \lambda\theta u_{i+1,j,k+1} - \lambda\theta u_{i,j-1,k+1} \\ &-\lambda\theta u_{i,j+1,k+1} = \lambda(1-\theta)u_{i-1,j,k} + \\ &(1-4\lambda(1-\theta))u_{i,j,k} + \lambda(1-\theta)u_{i+1,j,k} \\ &+\lambda(1-\theta)u_{i,j+1,k} + \lambda(1-\theta)u_{i,j-1,k} + \theta tF_{i,j,k+\frac{1}{2}} \end{aligned} \tag{2}$$

with $\theta = 0, \frac{1}{2}$ or 1.

1.2 Crack Propagation Problem

Fracture mechanics (Bui, 2006) is used to investigate the failure of brittle materials, which is to study material behavior and design against brittle failure and fatigue.

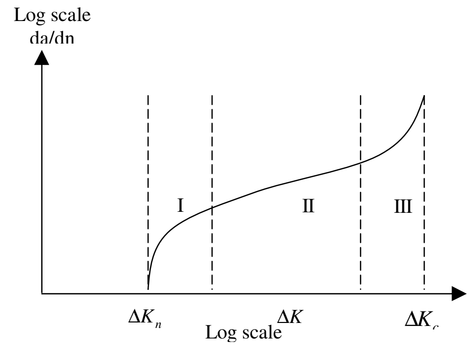


Figure 1: Failure of material view in three steps

The engineering study of fracture mechanics (Stanley, 1997) does not emphasize how a crack is initiated; the purpose is to develop methods of predicting how a crack propagates. Following are three steps failure of material: -

i. *Crack Initiation*: Initial crack occurs in this stage. This might be due to handling, tooling of the material, threads, and slip bands.
ii. *Crack Propagation*: During this stage, the crack continues growing as a result of continuously applied stresses.
iii. *Failure*: Failure occurs when the crack cannot withstand the applied stress on the material and happen quickly.

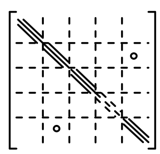
Adaptive finite element mesh is analysing two-dimension elastoplastic microfacture during crack propagation. Adaptive crack propagation calculation using finite element method is based on displacement formulation each element border.

2 Alternating Group Explicit (AGE)

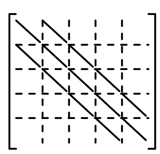
Based on the Douglas-Rachford formula, the AGE fractional scheme takes the form,

$$
\begin{aligned}
(G_1 + rI)u_{(r)}^{(k+\frac{1}{4})} &= (rI - G_1 - 2G_2 - 2G_3 - 2G_4)u_{(r)}^{(k)} + 2f \\
(G_2 + rI)u_{(r)}^{(k+\frac{1}{2})} &= G_2 u_{(r)}^{(k)} + ru_{(r)}^{(k+\frac{1}{4})} \\
(G_3 + rI)u_{(r)}^{(k+\frac{3}{4})} &= G_3 u_{(r)}^{(k)} + ru_{(r)}^{(k+\frac{1}{2})} \\
(G_4 + rI)u_{(r)}^{(k+1)} &= G_4 u_{(r)}^{(k)} + ru_{(r)}^{(k+\frac{3}{4})}
\end{aligned}
\tag{3}
$$

A is split into the sum of its constituent symmetric and positive definite matrices G_1, G_2, G_3, where,

$G_1+G_2=$  (4)

and

$G_3+G_4=$  (5)

with $diag(G_1+G_2) = diag(G_3+G_4) = \frac{1}{2}\,diag(A)$.

AGE fractional scheme is based on four intermediate levels, $(k+\frac{1}{4})$, $(k+\frac{1}{2})$, $(k+\frac{3}{4})$ and $(k+1)$. Using explicit (2×2) blocks for matrices (G_1+G_2) and (G_3+G_4), we have a group of (2×2) block systems which can be made explicit as follows,

$$
C_l = \begin{bmatrix}
r_l & & & & & & & \\
 & r_l & a_l & & & & & \\
 & a_l & r_l & & & & & \\
 & & & r_l & a_l & & & \\
 & & & a_l & r_l & & & \\
 & & & & & \ddots & & \\
 & & & & & & r_l & a_l \\
 & & & & & & a_l & r_l
\end{bmatrix}
\tag{6}
$$

3 Direct method of Crack Propagation

The initial step for modelling finite element started by elemental triangle of 6 nodes is used since it can fill in the most discrete finite element for dividing the model to small finite element, Ω_e. Composition of these elements will form a domain model, Ω,

$$\Omega = \sum_{e-1}^{n} \Omega_e \tag{7}$$

Consider that the extension of the element of a material is given by:

$$d\hat{u} = \frac{Fdx}{AE} \Rightarrow \frac{d\hat{u}}{dx} = \frac{F}{AE} \tag{8}$$

where $d\hat{u}$ is the extension of an element of length dx due to force, F. A and E are the Young's Modulus and constant cross sectional area of the element respectively. If F is constant over the element then $\frac{dF}{dx} = 0$. Hence equation (8) becomes:-

$$AE\frac{d^2\hat{u}}{dx^2} = 0 \tag{9}$$

Equation (9) is governing equation for an axial element. Integrating over the length l, get:-

$$AE\frac{d\hat{u}}{dx} = C_1 \tag{10}$$

$$AE\hat{u} = C_1 x + C_2 \tag{11}$$

Therefore,

$$\hat{u} = (u_j - u_i)\frac{x}{l} + u_i \tag{12}$$

After differentiation, we find from equation (12)

$$\frac{du}{dx} = \frac{(u_j - u_i)}{t} \tag{13}$$

at $x = x_i \rightarrow f_i = -AE\left(\frac{du}{dx}\right)_{x_i} = -AE\frac{(u_j - u_i)}{l}$

at $x = x_j \rightarrow f_j = AE\left(\frac{du}{dx}\right)_{x_j} = AE\frac{(u_j - u_i)}{l}$

The above can be expressed in matrix notation as

$$\begin{Bmatrix} f_i \\ f_i \end{Bmatrix} = \frac{AE}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} u_i \\ u_j \end{Bmatrix} \tag{14}$$

Finally, take

$$[K] = \frac{AE}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{15}$$

Then, for each element of domain can be written in,

$$[K_e]\{\Delta a_e\} = \{r_e\} \tag{16}$$

where,

$[K_e]$ = Elemental stiffness matrix,

$\{\Delta a\}$ = Constantly incremental displacement for border condition.

$\{r_e\}$ = Vector force including body force.

Elemental stiffness matrices tangent with all integration is carried out by using valuable gaussian integration technique. Elemental stiffness matrices is collected with standard finite element method to form global stiffness matrices,

$$[K] = \sum_{e=i}^{n} k_e \tag{17}$$

The problem solution is based on incremental iteration technique where the equation is non linear based on the law of elastoplastic material.

4 Parallelizing Strategies

4.2 Parallel AGE method

As the AGE method is fully explicit, its feature can be fully utilized for parallelization. Firstly, domain Ω is distributed to Ω^p sub domains by the master processor. The partitioning is based on domain decomposition technique.

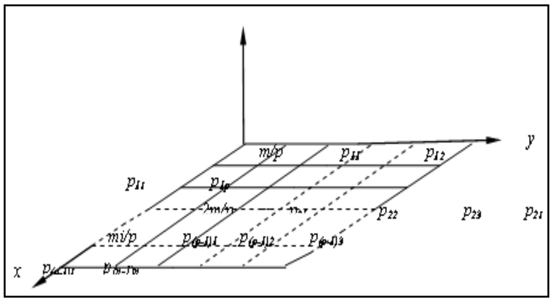


Figure 2: Domain decomposition for processor, $p_{i,j}$

Secondly, the sub domains Ω^p of AGE method is assigned into p processors in block ordering as illustrated in Figure 3. The domain decomposition for AGE method is implemented in five time levels. The communication activities between the slave processors are needed for the computations in the next iterations. The parallelization of AGE is achieved by assigning the explicit block (2×2). This proven that, the computations is involved are independent between processors. The parallelism strategy is straightforward with no overlapping sub domains. Based on the limited parallelism, this scheme can be effective in reducing computational complexity and data storage accesses in distributed parallel computer systems. The AGE sweeps involved tridiagonal systems, which in turn entails at each stage the solution of (2×2) block systems. The iterative procedure is continued until convergence is reached.

4.2 Parallel Direct method

In the master-worker model techniques, configuration of a central 'master' program communicates with a number of 'workers' (Krysl and Belyschko 1997). At the beginning of calculation, the master processor receives all the input data from user. Then all the data input are broadcasts from master processor node to all other processor node (Nikishkov and Kawka 1998). The system stiffness matrix is partitioned into a number of equally sized smaller domain matrices, each of which is allocated to a separate processor as shown in Figure 2. Each processor conducts part of duty that is required (Lewis and Hesham 1992). The domain matrices are solved independently. As soon as the calculation complete, domain matrices from each processor send to master processor for assembly to form global stiftness matrix as in Figure 3. When a slave completes one task, it requests another task from the master process (Wilkinson and Allen 1999).

5 Numerical Result

Numerical results of iterative and direct methods can be categorised into sequential and parallel algorithm using performance evaluation.

Figure 3 and 4 shows the result of direct method based on displacement and stress.

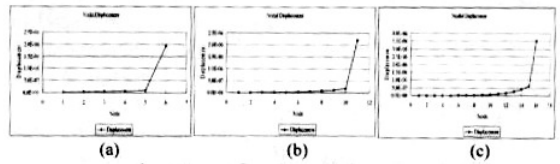


Figure 3: Displacement of each node for: (a) e=5 (b) e=10 (c) e=15

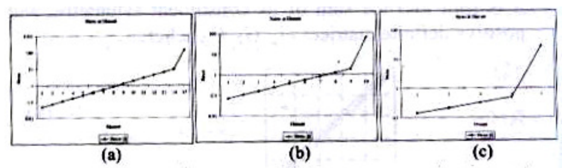


Figure 4: Comparison between stress at each element with: (a) e=5 (b) e=10 (c) e=15

5.1 Numerical Analysis

The following definitions are used to measure the parallel performance of the three methods,

Speed-up: $S_p = \frac{T_1}{T_p}$ and

Efficiency: $C_p = \frac{S_p}{p}$

Where T_1 is the execution time on one processor, T_p is the execution time on p processors. The important factors effecting the performance in message-passing paradigm on a distributed memory machine are communication patterns and computational/ communication ratios. Parallel algorithm for GSRB is chosen as a control scheme.

Effectiveness is used to make comparison among various parallel algorithms. The formula that is based on the calculation of speedup and the efficiency, given by the following,

$$Effectiveness = \frac{\text{Speedup}}{p.Time(t)}$$

5.2 Communication cost

The important factors affecting performance in message passing on distributed memory computer systems are communication patterns and computational or communication ratios. The communication time will depend on many factors including network structure and network contention. Parallel execution time t_{para} is divided into two parts, computational time (t_{comp}) and communication time (t_{comm}). t_{comp} is the time taken to compute arithmetic operations such as multiplication and addition operations in parallel algorithm. As all the processors doing the operation at the same speed, calculation for the t_{comm} is depending upon the size of message. The communication cost comes from two major phases in sending a message; start-up phase and data transmission phase. The total time to send K units of data for a given system can be written as,

$$t_{comm} = t_{startup} + K\, t_{data} + t_{idle}$$

where t_{comm} is time needed to communicate a message of K bytes. Sometimes, $t_{startup}$ is referred as the network latency time (start-up time), and a time to send a message with no data. It includes time to pack the message at source and unpack the message at the destination as well as to start a point-to-point communication. Meanwhile, t_{data} is time to transmit units of information; the reciprocal of t_d is the bandwidth. It is also the transmission time to send one bytes of data.

The $t_{startup}$ and t_{data} are assumed as constants and measured in bits/sec. t_{idle} is the time for message latency and waiting time for all processors to complete the works. The measurement of these communication costs are done via simple codes that extract the time during message exchange. The research focus on,

$$t_{parallel} = \text{time for parallel execution.}$$

$$t_{comm1} = \alpha\, t_{data} + \beta\, t_{startup}$$

where, α and β dependent on m and L.

Communication cost for parallel processing is:

$$L\, m\, t_{data} + L\,(t_{startup} + t_{idle})$$

Where m is units of data sending across processor and L is number of steps gained during overall execution.

6 Performance Evaluation and Discussion

In this work, we are using low cost cluster to execute parallel program. To solve this problem we use 6 to 20 numbers of processors to get numerical result and parallel performance evaluations.

Table 1 depicts the speedup and efficiency for parallel AGE, GSRB and direct methods. The speedup and efficiency values of AGE and direct methods get better as the size of processors increases which are consistent with the results in Figure 5 and 6. Comparable speedups are obtained for all applications with 6 processors. It can be observed that the efficiency of AGE method decreases faster than the direct method. This could be explained by the fact that several factors lead to increase in idle time such as network load, delay and load imbalance.

The stable and highly accurate AGE algorithms are found to be well suited for parallel implementation on the PVM platform where the data decomposition run asynchronously and concurrently at every time level. The AGE sweeps involve tridiagonal systems, which require the solution of (2×2) block systems.

	Speedup			Efficiency		
Pr	AGE	DIRECT	GSRB	AGE	DIRECT	GSRB
1	1	1	1	1	1	1.00000
2	1.916	1.9	1.70729	0.9577	0.95	0.85365
3	2.687	2.763356	2.39576	0.8957	0.921119	0.79859
4	3.439	3.475574	2.96449	0.8598	0.868893	0.74112
5	4.085	4.18518	3.47141	0.8170	0.81704	0.69428
6	4.737	4.83705	4.02634	0.7895	0.78951	0.67106

Table 1: Parallel performances of AGE, Direct and GSRB method

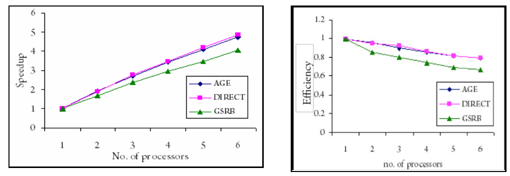


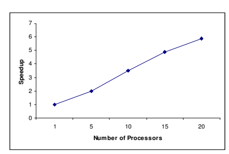
Figure 5: Speedup and efficiency vs. no. of processors

P	parallel	comp	ratio	comm	comm1	idle
5	31.995	27.645	6.36	4.3497	2.5472	1.8025
%		86.41		13.59	7.96	5.63
10	18.136	13.823	3.20	4.3133	2.5472	1.7661
%		76.22		23.78	14.04	9.74
15	12.987	9.2151	2.44	3.7719	2.5472	1.2247
%		70.96		29.04	19.61	9.43
20	10.75	6.9113	1.80	3.8387	2.5472	1.2915
%		64.29		35.71	23.69	12.01

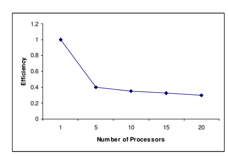
Table 2: Time execution, communication, idle times, ratio of computational time and communication time for the diffusion model problem.

p	parallel	comp	ratio	comm	comm1	idle
2	3310.85	3009.0	9.97	301.80	42.488	259.31
%		90.88		9.12	1.28	7.83
3	2276.44	2006.0	7.42	270.41	36.557	233.85
%		88.12		10.56	1.61	10.27
4	1709.95	1504.5	7.32	205.43	40.958	164.47
%		87.99		12.01	2.40	9.62

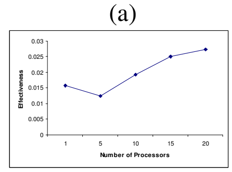
Table 3: Time execution, communication, idle times, ratio of computational time and communication time for the crack propagation problem.



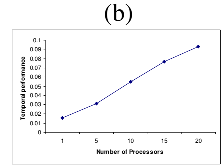
(a)



(b)



(c)



(d)

Figure 6: Parallel performance measurement for diffusion model problem in terms of (a) Speedup, (b) Efficiency, (c) Effectiveness and (d) Temporal performance.

It can be observed in Figure 5 that processors' number increment will result in higher rate of speedup. Unlike speedup, efficiency ratio will decrease as more processors added to the parallel system. Meanwhile, analysis in the aspect of effectiveness and temporal performance shows positive impact as processors' number increase.

7 Conclusions

In this work, we have presented the experimental results illustrating the parallel implementation of iterative block and direct method using PVM programming environment. The contribution of this paper is the parallelization of iterative and direct methods using block system is alternative to solve the large-sparse matrices system problem governing for finite difference and finite element methods.

Parallel algorithms for AGE and direct method are inherently explicit, the domain decomposition strategy is efficiently utilized and straightforward to implement on distributed memory architecture. Based on the analysis, it is noted that higher speedups could be expected for large-scale problems. Both block decomposition techniques and parallel implementations' advantages have fault tolerant features.

These schemes can be effective in reducing data storage accesses for computation and communication cost on a distributed computer systems.

The novelty of this paper is to implement of parallel iterative and direct methods in high performance evaluations on distributed memory architecture.

References

[1] Alias, N.: *Pembinaan dan Perlaksanaan Algoritma Selari bagi Kaedah TTHS dan TTKS dalam Penyelesaian Persamaan Parabolik pada Sistem Komputer Selari Ingatan Teragih.* Universiti Kebangsaan Malaysia. 2004.

[2] Alias, N., Sahimi, M.S., Abdullah, A.R.: *Parallel Strategies For The Iterative Alternating Decomposition Explicit Interpolation-Conjugate Gradient Method In solving Heat Conductor Equation On A Distributed Parallel Computer Systems*. Proceedings of The 3rd International Conference On Numerical Analysis in Engineering. 3: 31-38. 2003.

[3] B. Wilkinson and M. Allen. Parallel Programming Techniques and Apllications Using Networked Workstations and Parallel Computers. Prentice-Hall, 1999.

[4] Dahlquist, G., and Bjorck, A., 1974, Numerical Methods. Englewood Cliffs, N. Jersey: Prentice-Hall. 1974

[5] Evans, D.J., Sahimi, M.S.: *The Alternating Group Explicit Iterative Method (AGE) to Solve Parabolic and Hyperbolic Partial Differential Equations*. In Annual Review of Numerical Fluid Mechanics and Heat Transfer, Vol. 2. Hemisphere Publication Corporation, New York Washington Philadelphia London .1989.

[6] G.P.Nikishkov, M.Kawka, A.Makinouchi, G.Yagawa and S.Yoshimura, Porting an industrial sheet metal forming code to a distributed memory parallel computer. Computers and Structures , 67 pp. 439-449. 1998.

[7] Krysl P., Belytschko, T.:Modeling of 3D propagating cracks by the element-free Galerkin method, Proc. of Int. Conf. ICES'97, Costa Rica, 128-133. 1997.

[8] Ted Lewis and Hesham El-Rewini, Introduction to Parallel Programming, Prentice Hall, 1992.

[9] Tan, Y.S.: *Parallel Algorithm for One-dimensional Finite Element Discretization of Crack Propagation.* Universiti Teknologi Malaysia. 2008.